DECADE Problem Statement

draft-song-decade-problem-statement-01

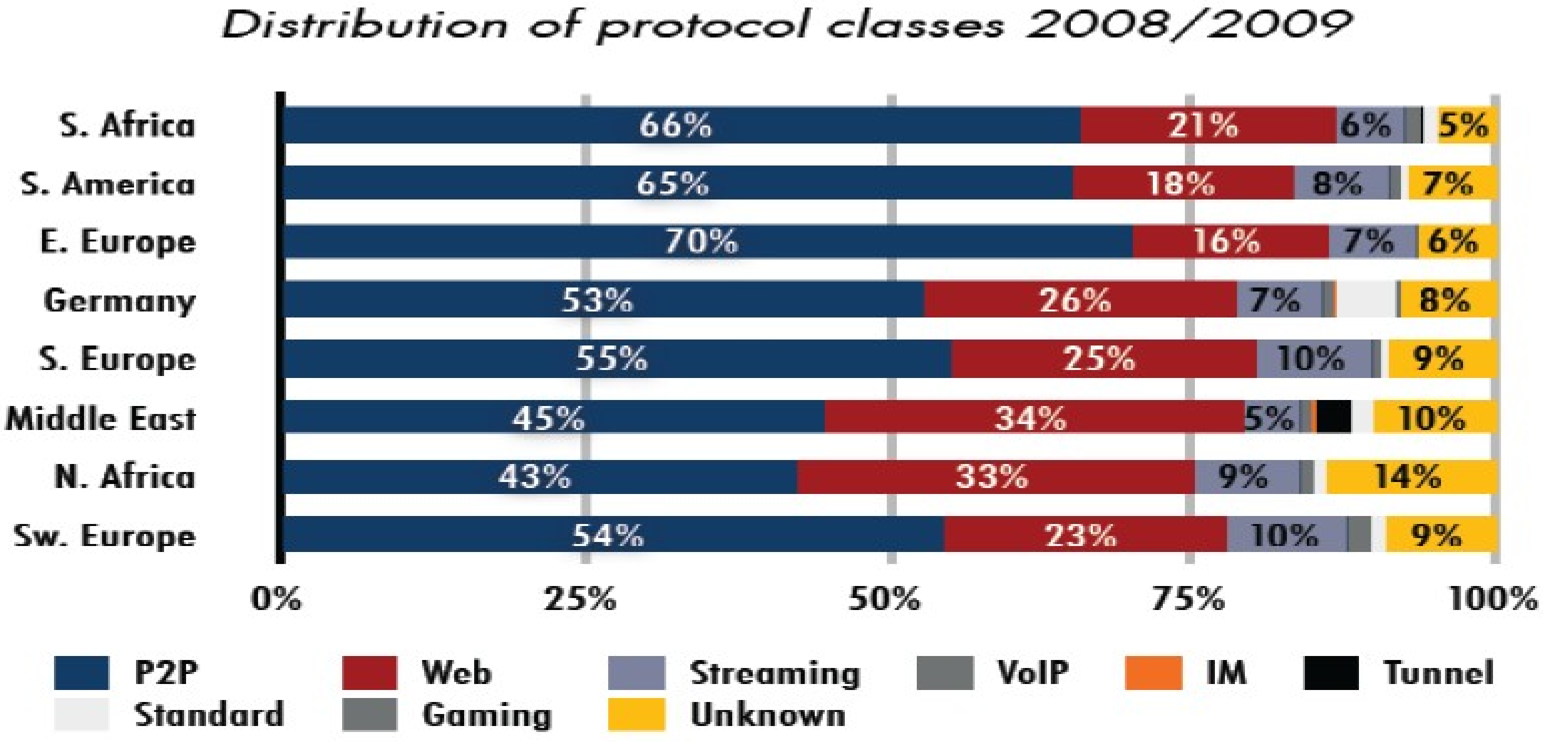
Richard Alimi
Haibin Song
Y. Richard Yang
Ning Zong

P2P Content Distribution Paradigm

- Highly-scalable
- Robust
- Space for innovation
    - Many novel techniques
    - Many players with novel ideas

P2P Contributes Significant Traffic

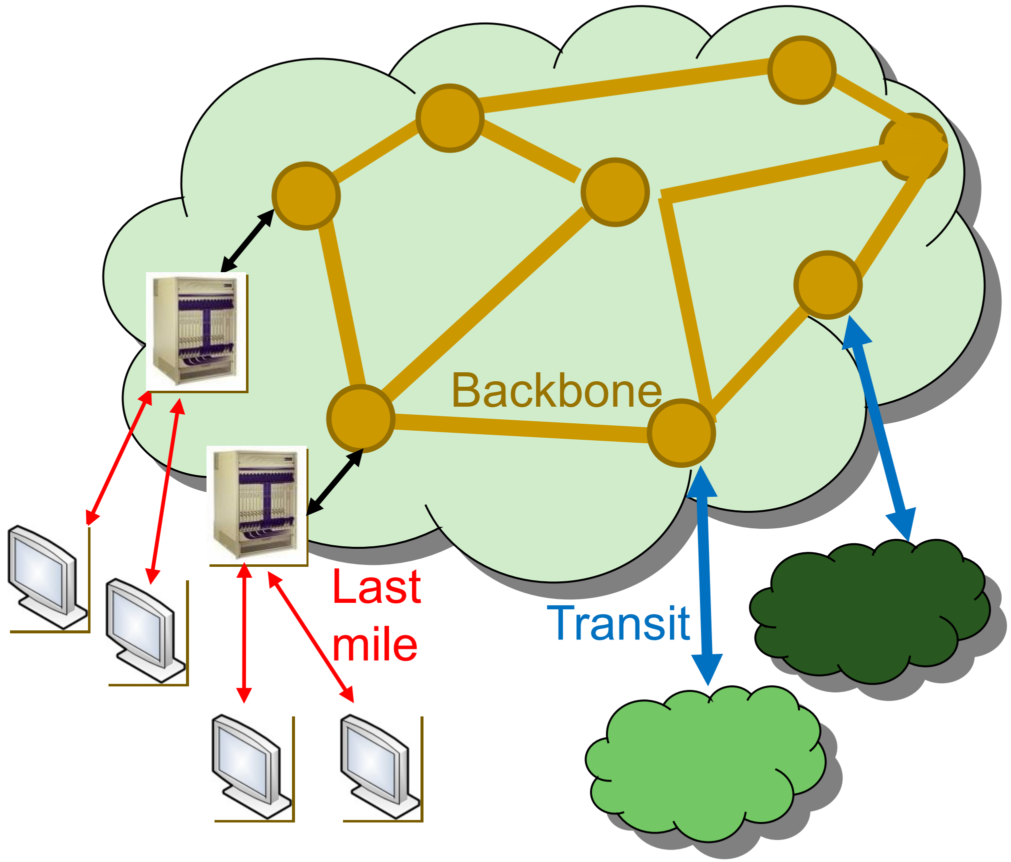
- 40-70% of total traffic in many networks



Source: ipoque Internet study 2008/2009

P2P Stress on Infrastructure

- Pure overlay distribution is inefficient
  - ❑ Transit
  - ❑ Backbone
  - ❑ Last mile



In-Network Storage

Effective technique to increase efficiency is to introduce *in-network storage*

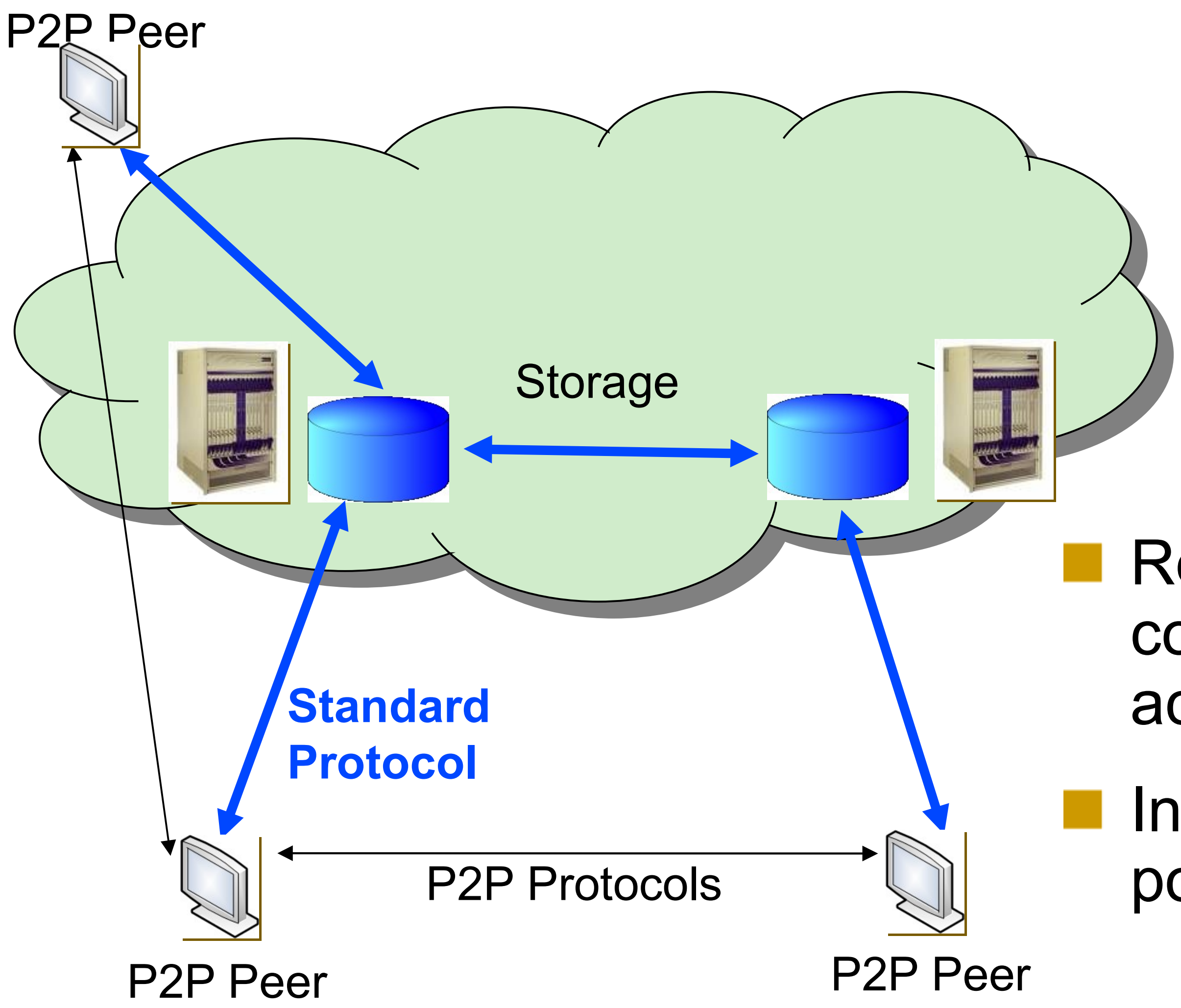
Problem 1: Weaknesses of Existing P2P Caches

- Tight coupling with P2P application protocol
  - *Cache must implement specific protocol for each application*
  - Large number of widely-used, evolving P2P protocols
    - File sharing: BitTorrent, eMule, Pando, …
    - Streaming: PPLive, PPStream, UUSee, Zattoo, Kontiki, TVAnts, Sopcast, Abacast, Solid State Networks, OctoShape, …
- *Implication*
  - *Cache vendor and ISP create and support complex production software*

Problem 2: Weak/No Integration with Applications

- Caches only consider policy from ISP perspective
  - *Application is out of the loop*
- *Implication*
  - *Application requirements/policies cannot be reflected*
  - Some P2P applications utilize resources (e.g., bandwidth) allocation amongst peers
  - Removing and refreshing stale content

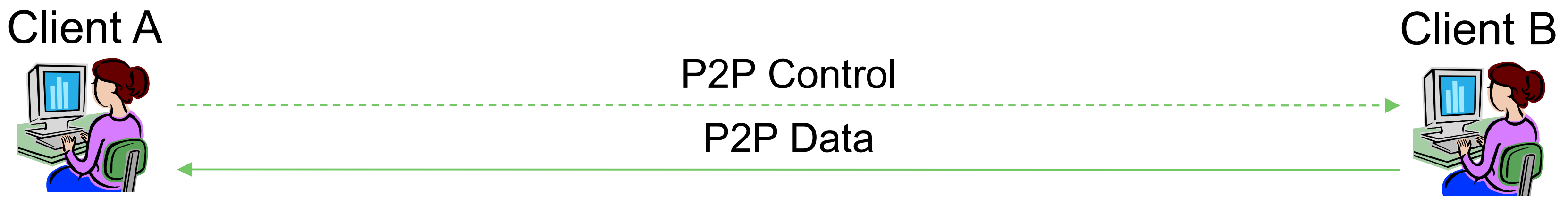
DECADE Overview



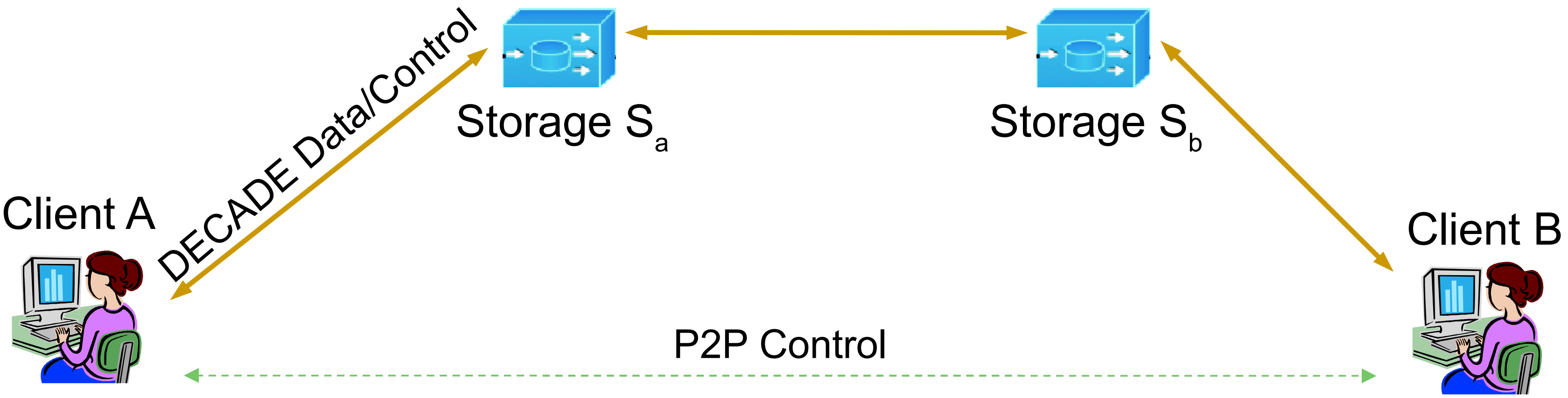
- Reduce production complexity and provide open access
- Integration with application policies

Example Operation

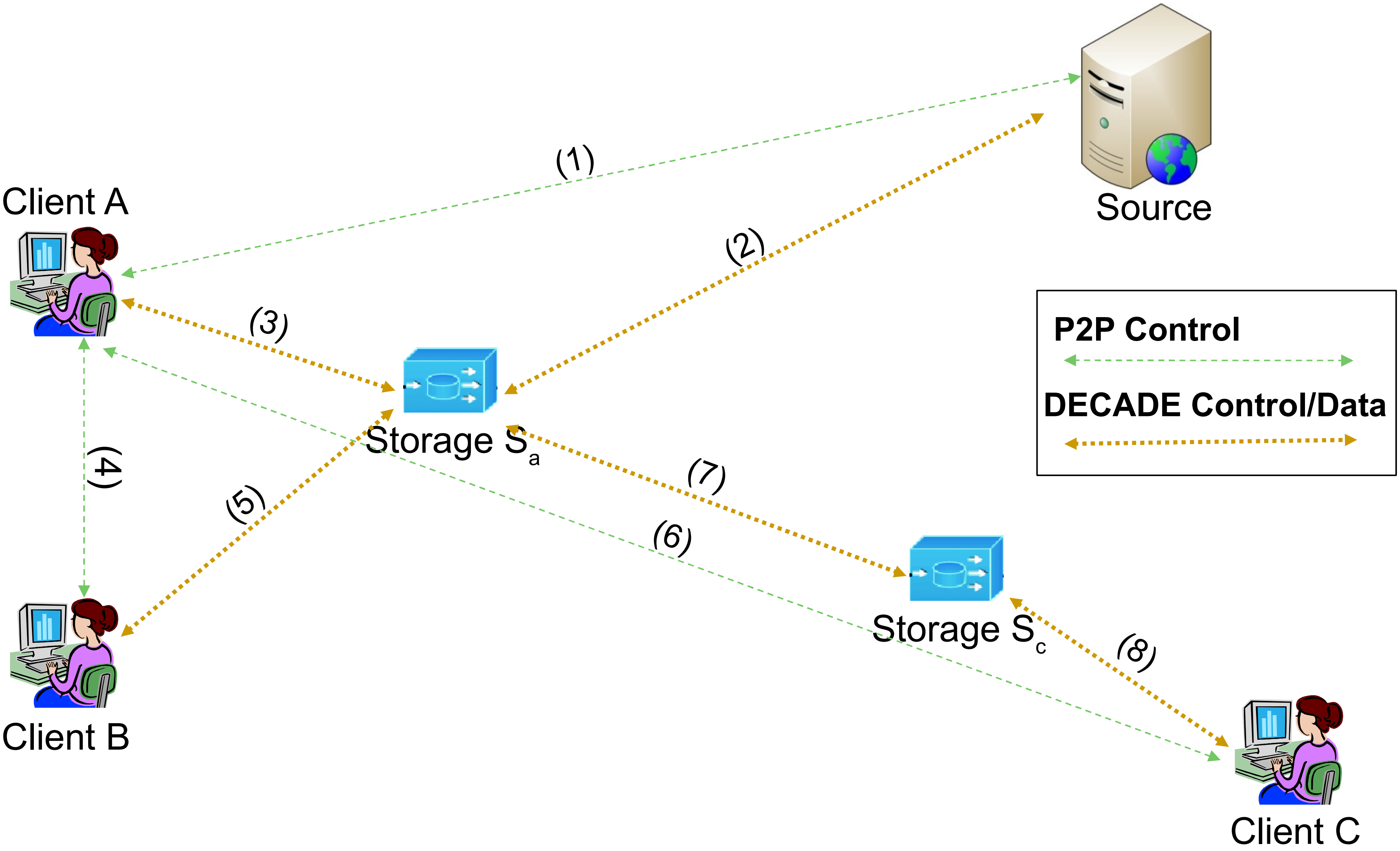
Native BitTorrent Clients



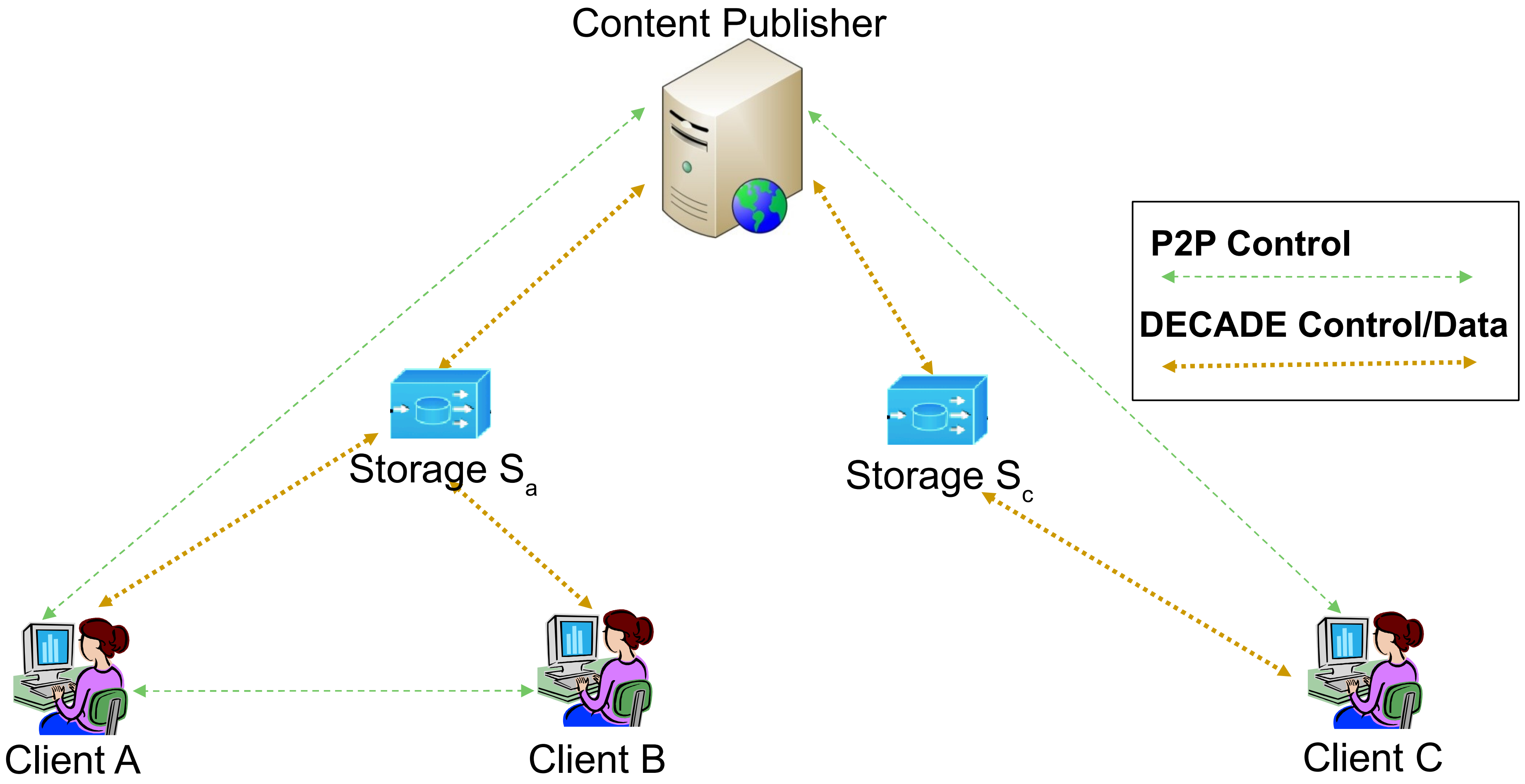
DECADE-enabled BitTorrent Clients



Use Case 1: P2P Users Sharing Content



Use Case 2: Content Publishers Sharing Content



Working Group Goal

Design standard protocol for diverse P2P applications to utilize in-network storage

Key Benefits

- Avoid complexity of existing P2P caching
- Allow integration with application policies
- Robustness
  - P2P applications still able to use to existing mechanisms
- Incremental deployment
- Open access to applications
- Open innovation for applications

Scope

- In-scope
  - A *standard, lightweight* control protocol between in-network storage and P2P applications
  - A *standard, lightweight* data-plane protocol for P2P applications to read from / write to in-network storage
  - Informational examples indicating how (one or two) existing P2P protocols can leverage in-network storage
- Out of scope
  - Standards indicating how specific P2P applications integrate with in-network storage

Comments and questions?

Key Components of In-network Storage

- Content Storage Mechanism
  - How are P2P contents detected and stored to in-network storage?
- Content Retrieval Mechanism
  - How are P2P contents discovered and read from in-network storage?
- Communcation Protocol
  - What is the protocol to communicate with in-network storage?

Existing Solution 1: Transparent P2P Cache

- Content Storage Mechanism
  - DPI detects content; content written to cache
- Content Retrieval Mechanism
  - Cache masquerades as peer
- Communcation Protocol
  - Existing P2P protocols

Existing Solution 2: Non-Transparent P2P Cache

- Content Storage Mechanism
  - Cache acts as a peer; content uploaded to it is cached
- Content Retrieval Mechanism
  - Cache acts as a peer; clients download as they would from any other peer
- Communcation Protocol
  - Existing P2P protocols